

**Screencast:** [25-mariadb-intro.webm](#) or [25-mariadb-intro.mp4](#)

## A brief history of MySQL and MariaDB

MySQL was originally created in 1995. The developers around it formed MySQL AB, a Swedish company, that mainly got its money from support contracts. Eventually MySQL became dual / multi-licensed with one of them being the GNU GPL.

In 2008 Sun Microsystems purchased MySQL AB for a reported \$1 billion although since that time many of the original / main developers have left Sun to form alternative projects. There is growing confusion as to who will end up being the driving force behind MySQL in the years to come. In January of 2010, Oracle acquired Sun Microsystems.

MySQL was designed to be a fast, light-weight database especially suited for the needs of web-based applications. Most often the M in "LAMP". MySQL is accessible from a large variety of programming languages including Perl, PHP, Python, C, C++ and from Java's ODBC.

MySQL has had a reputation of not being as feature complete as Oracle's database products and / or PostgreSQL. As a result some database purists turn their noses up at MySQL. As MySQL has matured many / most of the "missing features" have either been added or are on the roadmap for a future release.

The correct pronunciation of MySQL is "my es que el" and not "my sequel". SEQUEL was an IBM database language which was a predecessor to SQL. Unfortunately because of some other database products pronouncing SQL as "sequel" it might be the case that more people say it wrong than right.

If you want more than that, read the [MySQL wikipedia](#) page.

Some of the unique features of MySQL are:

1. Multiple database storage engines (table types) - InnoDB allows for transactions
2. Flexible table design - Easily insert columns and alter existing column data-types on-the-fly

**MariaDB** is a community-developed fork of the MySQL and development is led by some of the original MySQL developers. MariaDB was forked due to concerns MySQL's acquisition by Oracle Corporation. [Most Linux distributions provide mariadb.](#)



Monty Widenius and his two daughters, My and Maria

## Packages

mariadb-bench: MariaDB benchmark scripts and data  
 mariadb-libs: The shared libraries required for MariaDB clients  
 mariadb-server: The MariaDB server and related files  
 mariadb-test: The test suite distributed with MySQL  
 mariadb: MariaDB client programs and shared libraries

## Additional Packages

bytelfx-data-mysql: MySQL database connectivity for Mono  
 lua-sql-mysql: MySQL database connectivity for the Lua programming  
 mod\_auth\_mysql: Basic authentication for the Apache web server using a  
 MySQL-python: An interface to MySQL  
 pam\_mysql: PAM module for auth UNIX users using MySQL data base  
 perl-DBD-MySQL: A MySQL interface for perl  
 php-mysql: A module for PHP applications that use MySQL databases  
 qt3-MySQL: MySQL drivers for Qt 3's SQL classes  
 qt-mysql: MySQL driver for Qt's SQL classes  
 ruby-mysql: A Ruby interface to MySQL  
 tcl-mysqtlcl: MySQL interface for Tcl

## Files

Control script

`/usr/lib/systemd/system/mariadb.service`

Binaries

`/usr/bin/mysqld_safe` - Server application

`/usr/bin/mysql` - Client application (part of the mariadb package)

`/usr/bin/mysqldump` - Backup utility (part of the mariadb package)

### Configuration files

`/etc/my.cnf` (part of the mariadb-libs package)

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
# Disabling symbolic-links is recommended to prevent assorted
security risks
symbolic-links=0

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

Let's look at the files on the filesystem:

`cd /var/lib/mysql ; ls -lha` (Each directory is a database)

### Usage

Use a client to connect to the server and then run commands on the server.

Connect to mariadb-server using mysql text-based client:

```
mysql -u lee.dowdle -h host -p databasename
-p = prompt for password
databasename = set database to "use"
```

### Creating a database:

```
mysql> create database whatever;
```

### Choosing a database to use:

```
mysql> show databases;
mysql> use databasename;
mysql> show tables;
mysql> describe tablename;
```

### Creating a table:

See tutorial link below.

Example:

```
mysql> CREATE TABLE pet (name VARCHAR(20), owner VARCHAR(20),
-> species VARCHAR(20), sex CHAR(1), birth DATE, death DATE);
```

Give a user access to a database:

```
mysql> grant all on whatever.* to user@host identified by 'password';
```

Record actions:

INSERT, UPDATE, SELECT, DELETE

Please see MySQL's [tutorial](#).

## Backups

```
mysqldump -u root -p --all-databases --hex-blob --routines --triggers >  
mysql-complete-backup.sql
```

> = redirect output to a file since stdout is the default

You can get massive disk savings by compressing the .sql (120MB -> 25MB)

Of course you can do per-user backups, see man page for full details

What kind of file is an .sql file? A flat text file with .sql statements.

To import from an .sql file, just redirect from a file. Example:

```
mysql -u user -h hostname -p databasename < database.sql
```